Europäisches Patentamt
European Patent Office
Office européen des brevets

Bescheinigung Certificate Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr. Patent application No. Demande de brevet n°

02009133.6

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN COMPLIANCE WITH RULE 17.1(a) OR (b)

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

R C van Dijk

Europäisches Patentamt
European Patent Office
Office européen des brevets

Anmeldung Nr:
Application no.: 02009133.6
Demande no:

Anmeldetag:
Date of filing: 24.04.02
Date de dépôt:

Anmelder/Applicant(s)/Demandeur(s):

TELEFONAKTIEBOLAGET LM ERICSSON (publ)
Telefonvägen 30
126 25 Stockholm
SUEDE

Bezeichnung der Erfindung/Title of the invention/Titre de l'invention:
(Falls die Bezeichnung der Erfindung nicht angegeben ist, siehe Beschreibung.
If no title is shown please refer to the description.
Si aucun titre n'est indiqué se referer à la description.)

Tandem free operation in a communication network

In Anspruch genommene Priorität(en) / Priority(ies) claimed /Priorité(s) revendiquée(s)
Staat/Tag/Aktenzeichen/State/Date/File no./Pays/Date/Numéro de dépôt:

Internationale Patentklassifikation/International Patent Classification/
Classification internationale des brevets:

H04Q/

Am Anmeldetag benannte Vertragstaaten/Contracting states designated at date of filing/Etats contractants désignées lors du dépôt:

AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE TR

Tandem free Operation in a Communication Network

FIELD OF THE INVENTION

The present invention relates to the call set-up in a communication network. The invention especially relates to a method for set up a tandem free operation in a communication network for speech communication between a first communication terminal and a second communication terminal, whereby at least one of the terminals uses at least one codec type to encode the speech signals into an encoded data representation, with a first transcoder and a second transcoder wherein messages are send from the first transcoder to the second transcoder and vice versa to determine if both communication terminals have at least one codec type in common and if this is the case to establish a data connection between the first communication terminal and the second communication terminal without having the need to insert transcoding functions into the signal path between the first and the second communication terminal comprising the steps of exchanging messages between the transcoders that contain information on the encoder type currently used by the communication terminals, exchanging a second message between the transcoders as a response to the first message if both reported codec types match.

DESCRIPTION OF THE PRIOR ART

In digital systems for mobile communication speech signals are encoded by a speech encoder in order to reduce the data rate for saving bandwidth. In a normal call originating in a mobile station (MS) and terminating in a mobile station (MS), a so-called mobile to mobile call (MS-MS), the speech signal usually is encoded and decoded twice. In the originating mobile station the speech signal is encoded a first time before the encoded signal is sent over the air to a first base station A first transcoder decodes the encoded signal, which it receives from the first base station into a so-called a-law/μ-law signal which is commonly used in fixed communication networks. The decoded signal is routed in the fixed network to a second base station. Before the second base station can transmit the signal the signal is encoded

again in a second transcoder. The encoded signal is emitted by the second base station and is decoded in the terminating mobile station. The speech signal flow in the opposite direction is handled symmetrically.

As in this configuration two encoder/decoder pairs are lined up (the first speech encoder of the originating mobile station and the decoder of the originating transcoder is regarded as a first pair and the speech encoder of the terminating transcoder and the speech decoder of the terminating mobile station is regarded as a second pair) this configuration is called a speech codec "tandem". The key inconvenience of a tandem configuration is the speech quality degradation introduced by the double transcoding. This degradation is usually more noticeable when the speech codecs are operating at low transmission rates.

When the originating and terminating mobile stations are using the same type of speech codec, it is possible to transmit the speech frames received from the originating mobile station to the terminating mobile station without the need to activate the transcoding functions in the first and the second transcoder. As then there is only one pair of encoder and decoder (that is the encoder in the originating mobile station the decoder in the terminating mobile station) involved this configuration is called Tandem Free Operation (TFO). In modern networks, like UTMS, it is even possible to discard the whole Transcoder Hardware. This is then called a Transcoder Free Operation (TrFO). In TFO and TrFO mode the compressed speech signal is transmitted over the fixed network instead e.g. the usual a-law/μ-law signal. Besides the improvement of the speech quality by avoiding double transcoding this also saves costs as the compressed signal needs less bandwidth in the fixed network and power is saved since transcoding is bypassed. All necessary methods for negotiating, establishing and maintaining a Tandem Free Operating connection (TFO connection) are standardized for Codec types without configuration parameters (e.g. in GSM 08.62 for GSM_FR, GSM_HR and GSM_EFR) or are going to be standardized for more complex codec types (e.g.

the adaptive multi-code rate (AMR)) by the 3rd Generation Partnership Project (3GPP). In Technical Specification 3G TS 28.062 V.5.0.0; 3rd Generation Partnership project; Technical Specification Group Services & Systems Aspects; In-band Tandem Free Operation (TFO) of Speech Codecs; Stage 3- Service Description; Release 5 the content of which is to be incorporated here by reference all aspects of Tandem Free Operation for 3GPP is explained in detail.

Tandem Free Operation is activated and controlled in 3GPP by so-called Transcoder Units after the completion of the call set-up phase at both ends of a mobile-to-mobile call configuration. The TFO protocol is fully handled and terminated in the Transcoder Units. For this reason, the Transcoder Units cannot be bypassed in Tandem Free Operation. This is the key difference with the feature called Transcoder Free Operation (TrFO) defined in 3GPP TS 23.153. In return, the Transcoder Units continuously monitor the normal Tandem Free Operation and can terminate TFO as soon as necessary with limited impact on the speech quality. Before TFO is activated, the Transcoder Units exchange conventional 64 kbit/s PCM speech samples coded according to the ITU-T Recommendation G.711 [13] A-Law or m-Law. The Transcoders can also exchange TFO messages by stealing the least significant bit in every 16th speech sample (see annex A of 3GPP TS 28.062 V5.0.0 for the specification of the TFO message transmission rule and clauses 6 to 8 for the description of the TFO procedures and messages content). If compatible Speech Codec Types and Configurations are used at both ends of the mobile-to-mobile call configuration, the Transcoders automatically activate TFO. If incompatible Speech Codec Types and/or Configurations are used at both ends, then a codec mismatch situation exists. TFO cannot be activated until the codec mismatch is resolved. This capability is an optional feature involving other network elements of the Radio Access Network.

Once TFO is activated, the Transcoder Units exchange TFO Frames carrying compressed speech and in-band signaling, which structure is derived from the GSM

TRAU Frames defined in the 3GPP TS 48.060 and 48.061 (see clause 5). The exchange of TFO messages is still possible while TFO is active. In this case, the stealing process will result in embedding a message in the synchronization pattern of the TFO Frame.

The protocol flow shown in Fig. 1 is an example where immediate TFO setup is possible, either because both sides use identical Codec Types and Configurations, or because the Codec Types and/or Configurations are compatible in the "lower, contiguous subset". In the latter case potentially an optimisation phase might follow after TFO has been set up.

PROBLEM OF EXISTING TECHNOLOGY

If the transcoders find that the codec type currently used by both mobile terminals are compatible they will enter into a tandem free operation mode. Although the mobile terminals often support a codec type with better properties (e.g. better speech quality, better data rate) than the currently used codec type very often they will not start with the optimal codec type as in the described system only one codec type can be signaled. If both radio subsystems would always report the best codec supported by the respective mobile terminal no agreement will be found and the communication will be started in tandem mode, although a tandem free operation mode would have been possible. Although procedures ensure that in course of the communication the communication connection can be switched to a common codec type and thus tandem free operation can be enabled later on the signal distortions in tandem operation mode are inconvenient. Therefore a common strategy is to report a codec type with lower properties but that is widely spread in order to establish a tandem free operation very early. As further on messages (TFO_REQ_L; Con_Req frames) are exchanged that report a list of supported codec types of each terminal in the course of the connection the codec type may be changed to a better codec type.

However the users of the mobile terminals will experience the change of the codec as (click) noise and will be irritated.

The TFO standard version 5.0.0 was approved although other important decisions were still pending and although some shortcomings are known.
The major shortcomings in version 5.0.0 are:

No general TFO_Version handling is included in 3GPP TFO_Messages.
This means that no differentiation between older and newer TFO Versions is easily possible and leads to increased implementation effort. Some TFO_Version elements are included for some codec types, but this is not general enough and comes too late in the TFO_Protocol. A TFO_Version exchange exists in 3GPP2 standards (North America), but also this proposal sends the TFO Version too late (exactly when going into OPERATION) and it is not general enough (only a few bits reserved).

No sufficient extendibility for TFO_REQ is defined. There is a proposal on the table to indicate AMR-WB support when the AMR-NB is the Active Codec Type. This was, however, not accepted, because it was found to be not general enough. The discussion revealed that an extendibility for the existing TFO_REQ messages would be beneficial. Ideas are welcome. Other may also work on this. Our proposal is in Attachment 2.

No general solution exists to indicate support for Wideband speech (or any other alternative Codec Type) before the TFO is already in OPERATION in narrowband speech (the active Codec Type). Falling into TFO causes a slight speech distortion, falling out does this again. So it is desireable to know beforehand that a better Codec Alternative exists. Our proposal is also in Attachment 2.

No simple AMR-NB Configuration exchange. Recent discussions in 3GPP and Ericsson internal have the trend to simplify the AMR Configurations to a smaller

number of combinations. However, no simple solution for negotiating this has been developed

No simple AMR-WB Configuration exchange. The recent standardizations meetings discussed and approved such a substantial simplification for the AMR-WB Codec Family. However, also here no simple solution for negotiating this exists yet.

Another problem is that different protocol versions of the tandem free operation mode have to be supported as the protocol versions so far are not compatible.

Implementations according to TFO version 4.x may use two specific ways to transmit alternative Codec Types and Configuration parameters:

a) send TFO_REQ_L messages

b) send so called Config frames.

These config frames in release 4 can be embedded into No_Data frames (e.g. in speech pauses), into SID frames and into some of the various AMR Speech frames. The disadvantages are many:

1) several different kinds of configuration frames need to be implemented

2) these carry different amounts of configuration parameters (some do not carry all) and need separated handling

3) some do not have enough space to include the AMR-WB parameters

4) none has space enough for the next or future Codec Types

5) the configuration frames are all AMR specific and not codec-independent.

Therefore for Release 5 a new, "generic configuration frame" was proposed approved.

Therefore this has become state of the art. It simplifies implementations and allows future extensions. But: the TFO standard version 5.0.0 does not specify how interaction between version 4 and version 5 shall be handled. This is especially difficult as long as no TFO Version number is exchanged.

OBJECT OF THE INVENTION

It is therefore an object of the invention to allow very early in the call establishment a tandem free operation with prefetably the best common codec type. It is a further object of the invention to cope with the problem of incompatible protocol versions.

SUMMARY OF THE INVENTION

This problem is solved in that the first message contains further information on encoding capabilities of the respective communication terminal.

Some TFO_Message Extension_Blocks are defined together with their coding and usage. See Attachment 2. The intention is to generalise and simplify the TFO Mechanism for complex Codec Types such as AMR and AMR-WB and maintain high efficiency and low delay in the TFO Protocol.
For details see the attached word document. It contains the original TFO_Message definition (chapter 7 of the approved standard TS 28.062 Version 5.0.0) with additional revision marks for the new ideas.
Exactly this document (in its final form) will be presented as official 3GPP "Change Request" and - if approved - will be part of the standard, version 5.1.x.
For better understanding of the problems, please read the attached TFO-Standard TS 28.062 Version 5.0.0, especially section 6 and the Annex G.4.1 and Annex H.
Overview: Annex G.4.1 shows the typical example of a TFO Setup, when both sides use actually a compatible Codec Type and Codec Configuration. A reprint of this flow chart from the TFO standard is given on the next page, with the part under discussion marked with blue background. This is state of the art.
Argumentation: It is quite likely that in future releases of the GSM and UMTS networks speech calls will be setup with terminals and networks that support

besides several Narrow-Band Codec Types (such as GSM_FR, GSM_EFR, or AMR-NB) also the AMR-WB, with substantially improved speech quality due to the Wide-Band spectrum. But to get the advantage of this superior quality it is necessary that the whole path, both terminals and the network support this Codec Type.

In the early phase of deployment of this new Codec Type the likelihood that both end-terminals support this Codec Type is not very high. Thus typically the call will be started in NB quality, the TFO Protocol will then check whether WB is possible on both sides and finally the call will be transferred to WB quality.

With the currently standardised TFO Protocol-elements it is not possible to detect WB-capability before TFO is already set up. Therefore TFO must be terminated again, then the transfer to WB be done and finally TFO setup for the second time. This causes longer time until the wideband quality is available and it costs more distortions, because in general each TFO setup and release will cause a slight degradation (click or so).

The new TFO Protocol-elements will provide the necessary information on this alternative WB Codec Type already within the first TFO_REQ messages. Thus the TFO-Decision can be right the first time, saving time and distortion.

In one embodiment that further information relates to the version of the transcoder free protocol version that is supported by the transcoder.

In another embodiment that further information comprises a list of additional codec types that are alternatively supported by the respective communication terminal.

In another embodiment the further information comprises an indication that the message contains further information.

The new (claimed) extensions add information elements to the early exchanged TFO_REQ and TFO_ACK Messages and by that enable much more sophisticated decisions in the block "TFO Decision". Instead of going always into "Immediate TFO", these additional information elements allow the following variants:

No TFO at all, because the TFO_Version numbers or other configuration parameters reveal that the current call scenario is not favorable for TFO. This is important under several aspects and may gain even more importance in future evolutions of the TFO Standard.

One example for today: in TFO Protocol of Release 4.x.y so called "Configuration Parameters" are exchanged in a very specific and somehow scattered form within the Speech and No-Speech frames. The implementation of this needs some effort. Still, the solution is not flexible enough for extensions and the concept had to be given up when the AMR-WB was introduced in Release 5.0.0. Here a new concept for Configuration exchange was developed by the patenteethat overcomes this shortcomings. It is much more general, future proof and simpler to implement. But: to be backward compatible it is in general necessary to implement also the old mechanism. Now: with the early exchange of the TFO Version number it is possible that a Release 5 unit rejects the TFO attempt of a Release 4 unit and thus has a potential for cost saving.

No immediate TFO, because the new information elements show that a better TFO Configuration is possible. Instead Immediate Optimisation is performed: The TFO Protocol requests a change in Codec Type and Codec Configuration from call control and establishes TFO after that changes, e.g. after the transfer to AMR-WB. This in turn guarantees faster TFO Setup for the optimal configuration and better overall speech quality due to faster setup and less TFO establish/fallback cases.

The latter case was already described in one specialised example in standard-input document S4-020111.doc, see attachments. This document shows also in Annex G.9 the modified call flow. This is reprinted here on the next page. The new parts are marked with green background. It is already adapted a little bit by including the TFO Version (Ver) in the messages, although in this example the Version has no influence.

In another embodiment the version number is used in the receiving transcoder to look up a subset of an active codec set that is mandatory supported in each specific protocol version, wherby the transcoders compare that subset with the coder types supported by their mobile terminal and that the best codec type in common is chosen to enter into tandem free operation.

After a successful TFO Negotiation, when the protocol is in state OPERATION the old or new Configuration frames can be used. Now both sides know the TFO Versions and can decide how to handle the situation:

a) V4.x <-> V4.x: use old Configuration frames or the TFO_REQ_L and TFO_ACK_L protocol.

b) V4.x <-> V5.x: use old Configuration frames or the TFO_REQ_L and TFO_ACK_L protocol.

c) V5.x <-> V5.x: use new Configuration frames or

This in turn leads to a faster TFO setup in AMR-WB speech quality and should result in improved user satisfaction.

The TFO Messages are better prepared for further extensions.

The introduced TFO_Version negotiation allows deciding in an early stage of the TFO_Protocol, whether TFO setup is desirable or not. More implementation freedom for product design is the consequence.

BRIEF DESCRIPTION OF THE INVENTION

In the following the invention will be further described according to the figures and by means of examples

Fig.1: is a functional diagram of a communication system

Fig. 2 is a block diagram with the entities involved in a call between a GSM system and a 3GPP system

Fig. 3 is a protocol flow diagram illustrating the flow of messages .

Fig. 4 is a protocol flow diagram illustrating the flow of messages and the additional information included in the different messages.

EMBODIMENT OF THE INVENTION

FIG.1 shows a functional diagram of a system for mobile communication for exchanging speech signals between a first subscriber and a second subscriber. The communication system is shown in simplified form and is illustrative of a call routing mechanism between subscriber units within the communication system. Although the both subscriber units depicted in Fig. 1 are mobile terminals a subscriber unit also may be a fixed site terminal. However as fixed subscribers usually do not support a codec type tandem free operation is not possible in this case.

In the example depicted in Fig. 1 two mobile terminals are exchanging speech signals. For the sake of brevity, the communication system is illustrated and described with only limited amounts of infrastructure and subscriber equipment, although it will be readily appreciated that the system will comprise many

subscribers, base transceiver stations, TRAUs etc. for example, E.g. in some implementations additional switches may be used to freely assign TRAUs to different communication lines etc.

7 TFO Messages

The TFO Messages, introduced in clause 6, follow the generic IS_Message principle defined in annex A of 3GPP TS 28.062 V5.0.0.

The following definitions are provided for the ***Sender*** side:

TFO_REQ (): Identifies the source of the message as a TFO capable device, using a defined Codec_Type.

TFO_REQ contains the following parameters ():

- the System_Identification of the sender;
- the specific Local_Signature of the sender;
- the Local_Used_Codec_Type at sender side;
- possibly additional attributes for the Local_Used_Codec_Type.
- possibly additionally the TFO_Version
- possibly additionally alternative Codec_Types (short form of Codec_List)
- possibly additionally a future TFO_Extension

TFO_ACK (): Is the response to a TFO_REQ Message.

TFO_ACK contains the corresponding parameters as TFO_REQ, except for the Local_Signature replaced by the Reflected_Signature, copied from the received TFO_REQ Message.

TFO_REQ_L (): Is sent in case of Codec Mismatch or for sporadic updates of information.
TFO_REQ_L contains the following parameters ():

- the System_Identification of the sender;
- the specific Local_Signature of the sender;
- the Local_Used_Codec_Type at sender side;
- the Local_Codec_List of alternative Codec_Types;
- possibly additional attributes for the used and the alternative Codec_Types.

TFO_ACK_L (): Is the response to a TFO_REQ_L Message.
TFO_ACK_L contains the corresponding parameters as TFO_REQ_L, except for the Local_Signature replaced by the Reflected_Signature, copied from the received TFO_REQ_L Message.

TFO_TRANS (): Commands possible IPEs to let the TFO Frames pass transparently within the LSB (8 kbit/s) or the two LSBs (16 kbit/s) or the four LSBs (32kbit/s). TFO_TRANS contains the following parameter ():

- the Local_Channel_Type (8 kbit/s or 16 kbit/s or 32 kbit/s).

TFO_NORMAL: Commands possible IPEs to revert to normal operation.
TFO_NORMAL has no parameters.

TFO_DUP: Informs the distant partner that TFO Frames are received, while still transmitting PCM samples.
TFO_DUP has no parameters.

TFO_SYL: Informs the distant partner (if still possible) that TFO Frames are no longer received.
TFO_SYL has no parameters.

TFO_FILL: Message without specific meaning, used to pre-synchronise IPEs or to bridge over gaps in TFO protocols. TFO_FILL has no parameters.

7.1 EXTENDIBILITY

A mechanism for future extensions is defined in a way that existing implementations in the field shall be able to ignore future, for them unknown Codec_Types and their potential attributes. The existing implementations shall be able to decode the remainder of the messages (which is known to them) uncompromised. This mechanism allows to extent:

- the number of Local_Used_Codec_Types from 15 (short form) up to 255 (long form) for one System_Identification;
- the Codec_List;
- the Codec_Attributes (if needed).

In case of the TFO_REQ or TFO_ACK messages the attributes of the Local_Used_Codec_Type shall be sent in the codec specific way, without a preceding Codec_Attribute_Head Extension_Block. Existing equipment, that do not know a future Codec_Type and therefore do not know if and how many attribute Extension_Blocks do follow, shall skip these Extension_Blocks, until they find a TFO Message Header again. Similarly, if future Extension_Blocks to a known Codec_Type are detected, existing equipment shall skip these Extension_Blocks, until they find a TFO Message Header again.

In case of the TFO_REQ_L or TFO_ACK_L Messages the simple Codec_List shall be sent immediately after the SIG_LUC and possible Codec_x Extension_Blocks. Then the attributes of all alternative Codec_Types shall follow. Each set of codec attributes shall be preceded by the Codec_Attribute_Head Extension_Block (with

Codec_Type Identifier and Length Indicator) followed by the Codec specific attributes.

7.2 Regular and Embedded TFO Messages

A TFO Message is called "**regular**", if it is sent inserted into the PCM sample stream. A TFO Message is called "**embedded**", if it is embedded into a TFO Frame. The bit stealing scheme, as defined in Annex A, is identical for regular and embedded TFO Messages. The EMBED bit of the TFO Frames (see clause 5) indicates if the TFO Frame contains an embedded TFO Message. Due to the specific construction of the TFO Messages, they replace some of the synchronisation bits of the TFO Frames. Consequently, the TFO Frame synchronisation pattern will be affected by the presence of an embedded TFO Message, without compromising the synchronisation performances. Data and other control bits of the TFO Frames are not affected by embedded TFO Messages.

7.3 Cyclic Redundancy Check

The Extension_Blocks, defined in the following clauses, shall be protected by three CRC parity bits. These shall be generated as defined in the 3GPP TS 48.060 for the Enhanced Full Rate. For simplicity the present document is reprinted here:

"These parity bits are added to the bits of the subset, according to a degenerate (shortened) cyclic code using the generator polynomial:

$$g(D) = D^3 + D + 1$$

The encoding of the cyclic code is performed in a systematic form which means that, in GF(2), the polynomial:

$$d(m)D^n + d(m+1)D^{n-1} + \ldots\ldots + d(m+n-3)D^3 + p(0)D^2 + p(1)D + p(2)$$

where p(0), p(1), p(2) are the parity bits, when divided by g(D), yields a remainder equal to:

$$1 + D + D^2$$

For every CRC, the transmission order is p(0) first followed by p(1) and p(2) successively."

In case of Extension_Blocks, p(0)..p(2) are mapped to bits 16..18.

7.4 TFO_REQ MESSAGES

Symbolic Notation:

TFO_REQ (Sys_Id, LSig, Local_Used_Codec_Type[, Used_Codec_Attributes])

TFO_REQ_L (Sys_Id, LSig, Local_Used_Codec_Type, Codec_List [, Alternative_Codec_Attributes])

The TFO_REQ Messages conform to the IS_REQ Message format, defined in the Annex A, with IS_System_Identification, followed by the SIG_LUC Extension_Block, optionally the Codec_x Extension_Block, the Codec_List Extension_Block(s) and the Codec_Attribute Extension_Blocks.

The shortest TFO_REQ takes 140 ms for transmission, see Figure 7.4-1.

The shortest TFO_REQ_L takes 180 ms (Figure 7.4-2).

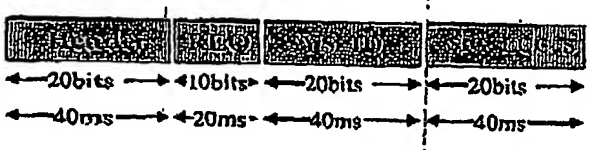

Figure 7.4-1: Construction of the shortest possible TFO_REQ Message

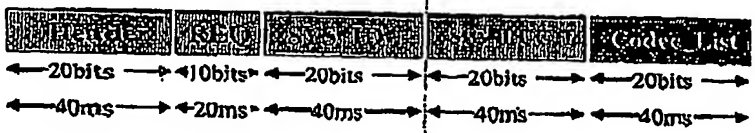

Figure 7.4-2: Construction of the shortest possible TFO_REQ_L Message

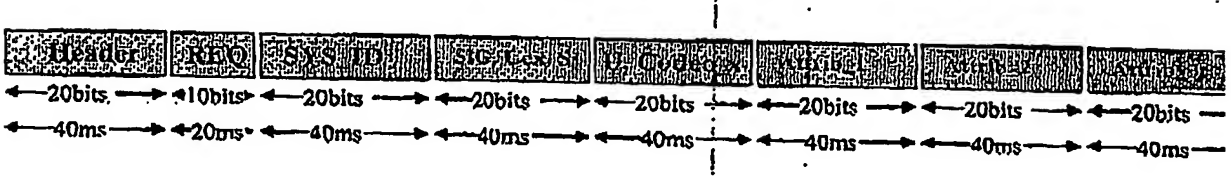

Figure 7.4-3: Example of a TFO_REQ Message with a Codec with an index higher than 15 and with three Attribute Extension_Blocks (300 ms length)

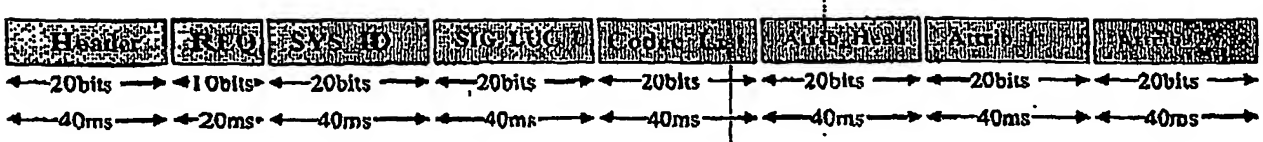

Figure 7.4-4: Example of a TFO_REQ_L Message with Codec_List and one alternative Codec with two Attribute Extension_Blocks (300 ms length)

A TFO_REQ (TFO_ACK) may have an additional TFO_Version Extension_Block that contains the TFO_Version.Subversion and a Selector. This Selector may indicate future extensions to TFO_REQ (TFO_ACK), which may require further additional Extension_Blocks following the TFO_Version, see figure 7.4-5.

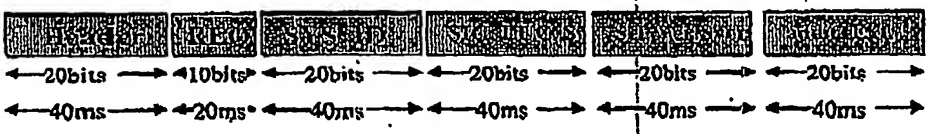

Figure 7.4-5: Construction of a TFO_REQ Message with Selector, TFO_Version.Subversion and one additional Extension_Block

7.4.1 Definition of the SIG_LUC Extension_Block

The SIG_LUC Extension_Block consists of 20 bits, as defined in Table 7.4.1-1. It shall always follow immediately after the SYS_ID Extension_Block. It differentiates a TFO_REQ from a TFO_REQ_L message and a TFO_ACK from a TFO_ACK_L message.

The Codec_x Extension_Block shall also be used in TFO_REQ or TFO_REQ_L messages if the Local_Used_Codec_Type has a CoID higher than 14.

Table 7.4.1-1: SIG_LUC Extension_Block

Bit	Description	Comment
Bit 1	"0"	normal IS-Message Sync Bit, constant.
Bit 2	List_Ind	Indicates, whether the Codec_List is included in the TFO Message or not 0: S: TFO_REQ or TFO_ACK: Codec_List is not included (short) 1: L: TFO_REQ_L or TFO_ACK_L: Codec_List is included (long)
Bit 3..10	Sig	An 8-bit random number to facilitate the detection of circuit loop back conditions an identify the message source
Bit 11	"0"	normal IS-Message Sync Bit, constant
Bit 12.. 15:	Codec_Type CoID_s (short form)	Identifies the Local_Used_Codec_Type, which is currently used by the sender 0000...1110: reserved for 15 Codec_Types 1111: Codec_x Extension_Block follows immediately
Bit 16..18:	CRC	3 CRC bits protecting Bits 2 to 10 and 12 to 15
Bit 19..20:	EX EX == "0.0" EX == "1.1"	The normal 2 bits for IS_Message Extension. No other extension block follows An other extension block follows

7.4.2 Definition of the Codec_x Extension_Block

The Codec_x Extension_Block, if present, always follows the SIG_LUC Extension_Block. It consists of 20 bits, as defined in Table 7.4.2-1. It shall follow always immediately after the SIG_LUC Extension_Block, if the Codec_Type field is set to "1111".

Table 7.4.2-1: Codec_x Extension_Block

Bit	Description	Comment
Bit 1	"0"	normal IS-Message Sync Bit, constant.
Bit 2	Codec_Sel	Differentiates the Codec_x Extension_Block 0: U: Used_Codec_Type is defined in Codec_Type_x field 1: Reserved
Bit 3..10	Codec_Type_x CoID (long form)	Identifies the Local_Used_Codec_Type, which is currently used by the sender 0000.0000 ... 1111.1111 reserved for 255 Codec_Types 0000.1111 is undefined and shall not be used.
Bit 11	"0"	normal IS-Message Sync Bit, constant
Bit 12.. 15:	"1010"	Reserved for future use, set to "1010" to minimise audible effects
Bit 16..18:	CRC	3 CRC bits protecting Bits 2 to 10 and 12 to 15
Bit 19..20:	EX	The normal 2 bits for IS_Message Extension. 00: No other extension block follows 11: An other extension block follows

7.4.3 Definition of the Codec_List_Extension_Block

The Codec_List Extension_Block is used in a TFO_REQ_L, TFO_ACK_L messages to list the supported Codec_Types. It consists of 20 bits, as defined in Table 7.4.3-1. The Codec_List must at least contain the Local_Used_Codec_Type.

If a system supports more than 12 Codec_Types, then other Codec_List Extension_Blocks (Table 7.4.3-2) may follow.

Table 7.4.3-1: Codec_List Extension Block

Bit	Description	Comment
Bit 1	"0"	Normal IS-Message Sync Bit, constant.
Bit 2..10	Codec_List_1	First part of Codec_List. For each Codec_Type one bit is reserved. If the bit is set to "0" then the specific Codec_Type is not supported; If the bit is set to "1" then the specific Codec_Type could be used.
Bit 11	"0"	Normal IS-Message Sync Bit, constant
Bit 12.. 14:	Codec_List_2	Second part of the Codec_List: All three bits are reserved for future Codec_Types (up to Codec_Type 12) :
Bit 15	Codec_List_x	If set to "1" a further Codec_List Extension_Block follows; otherwise set to "0"
Bit 16..18:	CRC	3 CRC bits protecting Bits 2 to 10 and 12 to 15
Bit 19..20:	EX	The normal 2 bits for IS_Message Extension: 00: No other extension block follows 11: An other extension block follows

Table 7.4.3-2: Further Codec_List Extension Block(s)

Bit	Description	Comment
Bit 1	"0"	normal IS-Message Sync Bit, constant.
Bit 2..10	Codec_List_1x	First part of Codec_List. For each Codec_Type one bit is reserved. If the bit is set to "0" then the specific Codec_Type is not supported; if the bit is set to "1" then the specific Codec_Type could be used. Bit 2: Codec_Type 13 (+ x*12; x=1..2..3) Bit 4: Codec_Type 14 (+ x*12; x=1..2..3) and so on
Bit 11	"0"	normal IS-Message Sync Bit, constant
Bit 12.. 14:	Codec_List_2x	Second part of the Codec_List: All three bits are reserved for future Codec_Types (up to Codec_Type 24 (+x*12; x=1..2..3)
Bit 15	Codec_List_xx	If set to "1" a further Codec_List Extension_Block follows; otherwise set to "0"
Bit 16..18:	CRC	3 CRC bits protecting Bits 2 to 10 and 12 to 15
Bit 19..20:	EX	The normal 2 bits for IS_Message Extension: 00: No other extension block follows 11: An other extension block follows

7.4.4 Definition of the Codec Attribute Head Extension Block

The Codec_Attribute_Head Extension_Block (Table 7.4.4-1) shall precede the Codec Attribute Extension_Blocks of a Codec_Type, if this Codec_Type needs additional attributes. This Codec_Attribute_Head identifies the Codec_Type and the number of additional Extension_Blocks to follow.

Table 7.4.4-1: Codec_Attribute_Head Extension_Block

Bit	Description	Comment
Bit 1	"0"	normal IS-Message Sync Bit, constant.
Bit 2	PAR_Sel	Differentiates this Extension_Block 0: Parameters included in PAR field: Simple Codec_List_Extension 1: Length indicator (LI) included: Parameters follow in subsequent Extension_Blocks
Bit 3..10	CoID	This field identifies the Codec_Type for which the subsequent attributes are valid. The same coding as in the Codec_x Extension_Block is used (long form)
Bit 11	"0"	normal IS-Message Sync Bit, constant
Bit 12.. 15:	LI / PAR	If Par_Sel==1: LI: Length indicator: 0000: reserved; 0001: one other Extension_Block follows; etc. If Par_Sel==0: PAR: Codec specific definition of these four bits
Bit 16..18:	CRC	3 CRC bits protecting Bits 2 to 10 and 12 to 15
Bit 19..20:	EX	The normal 2 bits for IS_Message Extension: 00: No other extension block follows 11: An other extension block follows

NOTE: This Extension_Block shall be used for the codecs introduced in the future that need attributes. It shall precede the Attribute Extension_Blocks. This allows earlier versions to skip the blocks they do not understand. It shall not be used for the GSM_FR, GSM_HR and GSM_EFR Codec_Types.

7.4.5 Definition of the TFO_Version Extension Block

The TFO_Version Extension_Block (Table 7.4.5-1) contains the "TFO_Version" (4 bit), the "TFO_Subversion" (4 bit) and a "Selector" (5bit). The TFO_Version Extension Block (and the additional Extension_Blocks indicated by the Selector, if any, see below) shall always be the last of Extension_Blocks of a TFO_REQ or TFO_REQ_L (or TFO_ACK or TFO_ACK_L) message. This is necessary to provide compatibility with older versions, which must be able to skip these Extension_Blocks without being effected negatively.

The TFO_Version and TFO_Subversion are specified in Annex H. A TFO implementation of Release 5 or onwards shall send this TFO_Version. If it is omitted then a TFO_Version lower than 5 shall be assumed by the receiving side.

The Selector is used to indicate the type of extension and the number of additional extension blocks (if any). The Selector code "00000" indicates that no further extension is following. The Selector code "10101" is not allowed to provide improved distinction against the TFO_Header.

Alternative Codecs

If the Selector is set to "00001" then this indicates that alternative codec types are supported, which are specified in additional Extension_Blocks following the TFO_Version Extension_Block. The Extension_Block shall not be used in TFO_REQ_L or TFO_ACK_L messages since equivalent information is provided then already by the Codec_List Extension_Block. It shall only be used in TFO_REQ or TFO_ACK messages to provide information on alternative codec types in an early stage of the TFO protocol, i.e., before TFO is established. For each alternative Codec_Type that is offered during TFO negotiation, one Codec_Attribute_Head Extension_Block shall be included. If the specified Codec_Type requires additional attributes then the required number of Codec_Attribute Extension_Blocks follow after the Codec_Attribute_Head Extension_Block. The list of alternative Codec_Types is terminated when the EX bits indicate no further Extension_Blocks (00) and the next TFO Message Header is following.

Table 7.4.5-1: TFO_Version Extension_Block

Bit	Description	Comment
Bit 1	"0"	normal IS-Message Sync Bit, constant.
Bit 2..6	Selector	Indicates if and which further extension_blocks are following. Coding for bits 2.3.4.5.6: 00000: nothing is following after this TFO_Version 00001: One (or more) alternative Codec Type(s) is (are) following, 10101: reserved (used by the IS_Header) all other codes: reserved for future use.
Bit 7..10	Ver	This field contains the TFO_Version number as specified in Annex H
Bit 11	"0"	normal IS-Message Sync Bit, constant
Bit 12.. 15:	Sver	This field contains the TFO_Subversion number as specified in Annex H
Bit 16..18:	CRC	3 CRC bits protecting Bits 2 to 10 and 12 to 15
Bit 19..20:	EX	The normal 2 bits for IS_Message Extension: 00: No other extension block follows 11: An other extension block follows

7.5 TFO_ACK Messages

Symbolic Notation:

TFO_ACK(Sys_Id, RSig, Local_Used_Codec_Type [, Used_Codec_Attributes])

TFO_ACK_L (Sys_Id, RSig, Local_Used_Codec_Type, Codec_List [, Alternative_Codec_Attributes]).

The TFO_ACK Messages conform to the IS_ACK Message, defined in the Annex A, with IS_System_Identification, followed by the SIG_LUC Extension_Block, and optionally the Codec_x Extension_Block, the Codec_List Extension_Block(s) and the Codec_Attribute Extension_Blocks.

TFO_ACK and TFO_REQ Messages differ only in the ACK / REQ Command block and the construction of the Signature: Local_Signature in case of TFO_REQ, Reflected_Signature in case of TFO_ACK. All extension blocks defined for the TFO_REQ are valid as well for TFO_ACK.

The shortest TFO_ACK takes 140 ms for transmission.

The shortest TFO_ACK_L takes 180 ms.

7.6 TFO_TRANS Messages

Symbolic Notation: TFO_TRANS (Channel_Type).

Two TFO_TRANS Messages are defined in conformity to the IS_TRANS Messages in Annex A of 3GPP TS 28.062 V5.0.0.

For 8 kbit/s submultiplexing the "TFO_TRANS (8k)" is used and is identical to "IS_TRANS_1_u".

For 16 kbit/s submultiplexing the "TFO_TRANS (16k)" is used and is identical to "IS_TRANS_2_u".

For 32 kbit/s submultiplexing the "TFO_TRANS (32k)" is used and is identical to "IS_TRANS_4_u".

TFO_TRANS() takes 100 ms for transmission.

In most cases the respective TFO_TRANS Message shall be sent twice: once as a regular TFO Message, exactly before any series of TFO Frames, and once embedded into the first TFO Frames, see clause 10.

7.7 TFO_NORMAL MESSAGE

Symbolic Notation: TFO_NORMAL.

The TFO_NORMAL Message is identical to the IS_NORMAL Message defined in the Annex A of 3GPP TS 28.062 V5.0.0

It shall be sent at least once whenever an established Tandem Free Operation needs to be terminated in a controlled way.

TFO_NORMAL takes 100 ms for transmission.

7.8 TFO_FILL MESSAGE

Symbolic Notation: TFO_FILL.

The TFO_FILL Message is identical to the IS_FILL Message, defined in the Annex A of 3GPP TS 28.062 V5.0.0.

TFO_FILL may be used to pre-synchronise IPEs. Since IS_FILL is one of the shortest IS Messages, this is the fastest way to synchronise IPEs, without IPEs swallowing other protocol elements. By default three TFO_FILL messages shall be sent at the beginning; this number may be, however, configuration dependent.

One TFO_FILL takes 60 ms for transmission.

7.9 TFO_DUP MESSAGE

Symbolic Notation: TFO_DUP

The TFO_DUP Message is identical to the IS_DUP Message, defined in Annex A. TFO_DUP informs the distant TFO Partner, that TFO Frames have been received unexpected, e.g. during Establishment. This enables a fast re-establishment of TFO after a ***local*** handover.

TFO_DUP takes 60 ms for transmission.

7.10 TFO_SYL MESSAGE

Symbolic Notation: TFO_SYL

The TFO_SYL Message is identical to the IS_SYL Message, defined in Annex A. TFO_SYL informs the distant TFO Partner, that tandem free operation has existed, but suddenly no TFO Frames were received anymore. This enables a fast re-establishment of TFO after a ***distant*** handover.

TFO_SYL takes 60 ms for transmission.

7.11 SPECIFICATION OF THE TFO MESSAGES

7.11.1 CODEC TYPES

The Codec_Types are defined according to 3GPP TS 26.103, table 6.3-1.

The short form (CoID_s) exists for all Codec_Types with indices below 15 and consists of the last four bits (LSBs) of the long form (CoID).

7.11.2 CODEC LIST

The Codec_List is defined according to 3GPP TS 26.103. The mapping into the Codec_List Extension block shall be as follows: bit 1 of octet 1 shall be placed into Bit 2 of the Codec_List Extension block, and so on until bit 4 of octet 2 shall be placed into Bit 14.

If more than 12 Codec Types are contained in the Codec_List, then Bit 15 of the first Codec_List Extension block shall be set to "1" and an further Codec_List Extension block shall be added for the next 12 Codec Types.

7.11.3 Codec_Type Attributes

The Codec_Types GSM Full Rate, GSM Half Rate and GSM Enhanced Full Rate do not need additional attributes. They are fully defined by the System_Identification (see Annex A.5) and the Codec_Type.

7.11.3.1 AMR Codec_Type Attributes

The Adaptive Multi-Rate Codec_Types (FR_AMR, HR_AMR, UMTS_AMR, UMTS_AMR_2) and the Adaptive Multi-Rate Wideband Codec_Types (FR_AMR-WB and UMTS_AMR-WB) need several attributes within the TFO_REQ and TFO_ACK as well as in the TFO_REQ_L and TFO_ACK_L Messages. For Con_Req and Con_Ack frames see Annex C.

There are two major kinds of attributes: the ACS (Active Codec Set) and potentially the SCS (Supported Codec Set)The ACS is related to the Local_Used_Codec_Type and is part of the Used_Codec_Attributes. One and exactly one ACS shall be sent in all cases where the Local_Used_Codec_Type is FR_AMR, HR_AMR, UMTS_AMR, or UMTS_AMR_2, FR_AMR-WB or UMTS_AMR-WB within one ACS_Extension_Block. This ACS_Extension_Block carries some more parameters as defined in the next clause, the most important one is the "Full_Sub" flag, indicating whether or not the full set or a sub-set of the AMR (AMR-WB) is supported. In TFO_REQ and TFO_ACK Messages the ACS shall follow immediately after the SIG_LUC_Extension_Block. In TFO_REQ_L and TFO_ACK_L Messages an Attribute_Head_Extension_Block shall follow after the Local_Codec_List, indicating the Codec_Type it specifies, followed by the corresponding ACS_Extension_Block.

The SCS shall be sent in TFO_REQ or TFO_ACK only if the ACS_Extension_Block indicates that the sending side does not support the full set of AMR codec modes, but a subset (Full_Sub flag). In this case the SCS_Extension_Block shall follow immediately after the ACS_Extension_Block.

NOTE 1: Hence, the TFO_Protocol can decide immediately after the reception of TFO_REQ or TFO_ACK whether TFO is possible or not, and can report the distant TFO parameters to the Control Entity in the Network.

One and only one ACS_Extension_Block is included in TFO_REQ_L and TFO_ACK_L, if the Local_Used_Codec_Type is FR_AMR, HR_AMR, UMTS_AMR or UMTS_AMR_2, FR_AMR-WB or UMTS_AMR-WB. In addition one SCS_Extension_Block is needed for each AMR Codec_Type flagged in the Local_Codec_List. In that case an Attribute_Head_Extension_Block shall follow after the Local_Codec_List, indicating the Codec_Type it specifies, followed by the corresponding SCS_Extension_Block. If multiple AMR_Codec_Types are flagged, then multiple Attribute_Heads and SCS_Extension_Blocks may be needed. If the full set of AMR Codec Modes is supported, then neither the Attribute_Head nor the SCS_Extension_Block shall be sent for the alternative Codec_Type(s).
The following figures give the examples for the full-set AMR TFO Messages.

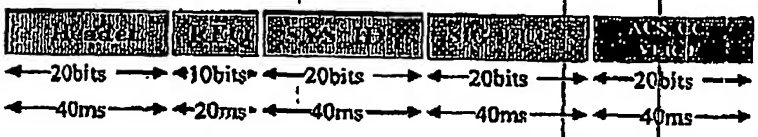

Figure 7.11.3.1-1: Construction of the shortest possible TFO_REQ Message for any AMR Codec Type

TFO_ACK follows the same construction. Both have a length of 180ms.

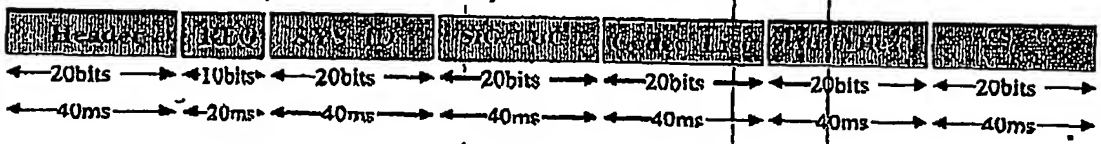

Figure 7.11.3.1-2: Construction of the shortest possible TFO_REQ_L Message listing an AMR Codec_Type in the Codec_List

TFO_ACK_L follows the same construction. Both have a length of 260ms.

NOTE 2: In TFO_REQ_L (TFO_ACK_L) at least one Attribute_Head is needed, if the Local_Used_Codec_Type is AMR or AMR-WB, because otherwise a TFO partner that does not know the Local_Used_Codec_Type cannot know how many attributes are needed – if any. Since these longer messages are used only when mismatch is identified or in other situations, where protocol speed is not important, this additional 40ms message length is not important.

In the worst case in GSM, when both AMR Codec_Types and the FR_AMR-WB are flagged in the Codec_List, but none supports the full set, then seven Extention_Blocks need to follow after the Codec_List.

Example: FR_AMR == Local_Used_Codec_Type: Attribute_Head(FR_AMR) – ACS(FR_AMR) – SCS(FR_AMR) – Attribute_Head(HR_AMR) – SCS(HR_AMR) – Attribute_Head(FR_AMR-WB) – SCS(FR_AMR-WB)

7.11.3.1.1 AMR ACTIVE CODEC SET ATTRIBUTES

One AMR_ACS (AMR-WB_ACS) Extension_Block shall be added in the TFO_REQ and TFO_ACK messages after the SIG_LUC Extension_Block if an AMR (AMR-WB) Codec_Type is used as the Local_Used_Codec_Type.

Table 7.11.3.1.1-1: AMR_ACS Extension_Block

Bit	Description	Comment
Bit 1	"0"	Normal IS-Message Sync Bit, constant.
Bit 2..9	Active Codec Set (NB_ACS)	Active Codec Set: For each Codec_Mode of the AMR one bit is reserved. If the bit is set to "0" then the specific Codec_Mode is not in the ACS, otherwise it is in and may be used by the adaptation algorithm. Bit 2: AMR_Mode 12,2 kbit/s (undefined for HR_AMR) Bit 3: AMR_Mode 10,2 kbit/s (undefined for HR_AMR) Bit 4: AMR_Mode 7,95 kbit/s Bit 5: AMR_Mode 7,40 kbit/s Bit 6: AMR_Mode 6,70 kbit/s Bit 7: AMR_Mode 5,90 kbit/s Bit 8: AMR_Mode 5,15 kbit/s Bit 9: AMR_Mode 4,75 kbit/s
Bit 10	Full_Sub (NB_F/S)	0: Full Set supported, NB_SCS is not following 1: Subset only supported, NB_SCS is following immediately
Bit 11	"0"	Normal IS-Message Sync Bit, constant
Bit 12	spare	set to "1"
Bit 13	Optimisation Mode (NB_OM)	ACS Optimisation Mode 0 No ACS Change supported 1 ACS change supported
Bit 14 & 15	NB_Ver	Version Number of the AMR-NB TFO Scheme Bit 15 is equivalent to the ATVN in Configuration Frames, see Annex C
Bit 16..18	CRC	3 CRC bits protecting Bits 2 to 10 and 12 to 15
Bit 19..20:	EX	The normal 2 bits for IS_Message Extension: 00: No other extension block follows 11: An other extension block follows (i.e. SCS)

Table 7.11.3.1.1-2: AMR-WB_ACS Extension_Block

Bit	Description	Comment
Bit 1	"0"	Normal IS-Message Sync Bit, constant
Bit 2..10	Active Codec Set (WB_ACS)	Active Codec Set: For each Codec_Mode of the AMR-WB one bit is reserved. If the bit is set to "0" then the specific Codec_Mode is not in the ACS, otherwise it is in and may be used by the adaptation algorithm. Bit 2: AMR-WB_Mode 23.85 kbit/s Bit 3: AMR-WB_Mode 23.05 kbit/s Bit 4: AMR-WB_Mode 19.85 kbit/s Bit 5: AMR-WB_Mode 18.25 kbit/s Bit 6: AMR-WB_Mode 15.85 kbit/s Bit 7: AMR-WB_Mode 14.25 kbit/s Bit 8: AMR-WB_Mode 12.65 kbit/s Bit 9: AMR-WB_Mode 8.85 kbit/s Bit 10: AMR-WB_Mode 6.60 kbit/s
Bit 11	"0"	Normal IS-Message Sync Bit, constant
Bit 12	Full_Sub (WB_F/S)	0: Full Set supported, WB_SCS is not following. 1: Subset only supported, WB_SCS is following immediately
Bit 13	Optimisation Mode (WB_OM)	ACS Optimisation Mode 0: No ACS Change supported 1: ACS Change supported
Bit 14	spare	set to "1"
Bit 15	spare	set to "1"
Bit 16..18	CRC	3 CRC bits protecting Bits 2 to 10 and 12 to 15
Bit 19..20:	EX	The normal 2 bits for IS_Message Extension: 00: No other extension block follows 11: An other extension block follows (i.e. SCS)

7.11.3.1.2 AMR SUPPORTED CODEC SET ATTRIBUTES

The AMR_SCS (AMR-WB_SCS) Extension_Block contains the information on the AMR (AMR-WB) Supported Codec Set. It shall be omitted, if the full set is supported. Table 7.11.3.1.2-1 gives the description of the SCS Extension_Block. For the Local_Used_Codec_Type the SCS Extension_Block shall follow immediately after the corresponding ACS Extension_Block. In that case the Full_Sub flag shall be set within the ACS Extension_Block. For alternative Codec_Types, as flagged in the Local_Codec_List, the SCS shall follow immediately after the corresponding Attribute_Head Extension_Block.

NOTE: The VERsion numbers in ACS and SCS Extension_Blocks shall be identical for one Codec_Type, but may be different for different Codec_Types (e.g. FR_AMR and HR_AMR or FR_AMR-WB).

Table 7.11.3.1.2-1: AMR_SCS Extension_Block

Bit	Description	Comment
Bit 1	"0"	Normal IS-Message Sync Bit, constant.
Bit 2...9	Supported Codec Set (NB_SCS)	Supported Codec Set: For each Codec_Mode of the AMR one bit is reserved. If the bit is set to "0" then the specific Codec_Mode is not supported; if the bit is set to "1" then the specific Codec_Mode is supported and may be considered for the optimisation of the common ACS. Bit 2: AMR_Mode 12,2 kbit/s (undefined in SCS(H)) Bit 3: AMR_Mode 10,2 kbit/s (undefined in SCS(H)) Bit 4: AMR_Mode 7,95 kbit/s Bit 5: AMR_Mode 7,4 kbit/s Bit 6: AMR_Mode 6,7 kbit/s Bit 7: AMR_Mode 5,9 kbit/s Bit 8: AMR_Mode 5,15 kbit/s Bit 9: AMR_Mode 4,75 kbit/s
Bit 10	NB_MACS MSB	See comment for Bit 12...13
Bit 11	"0"	normal IS-Message Sync Bit, constant
Bit 12...13	NB_MACS LSBs	The maximally supported number of Codec_Modes in this radio leg. Coding for bits 10.12.13: "0.0.1" 1 Mode "0.1.0" 2 Modes "0.1.1" 3 Modes "1.0.0" 4 Modes "1.0.1" 5 Modes "1.1.0" 6 Modes "1.1.1" 7 Modes "0.0.0" 8 Modes
Bit 14...15	NB_Ver	Version Number of the AMR TFO Scheme for that Codec_Type Bit 15 is equivalent to the ATVN in Configuration Frames, see Annex C
Bit 16..18	CRC	3 CRC bits protecting Bits 2 to 10 and 12 to 15
Bit 19 20	EX	The normal 2 bits for IS_Message Extension: 00: No other extension block follows 11: An other extension block follows

Table 7.11.3.1.2-2: AMR-WB_SCS Extension_Block

Bit	Description	Comment
Bit 1	"0"	Normal IS-Message Sync Bit, constant.
Bit 2...10	Supported Codec Set (WB_SCS)	Supported Codec Set: For each Codec_Mode of the AMR-WB one bit is reserved. If the bit is set to "0" then the specific Codec_Mode is not supported; if the bit is set to "1" then the specific Codec_Mode is supported and may be considered for the optimisation of the common WB_ACS. Bit 2: AMR-WB_Mode 23.85 kbit/s Bit 3: AMR-WB_Mode 23.05 kbit/s Bit 4: AMR-WB_Mode 19.85 kbit/s Bit 5: AMR-WB_Mode 18.25 kbit/s Bit 6: AMR-WB_Mode 15.85 kbit/s Bit 7: AMR-WB_Mode 14.25 kbit/s Bit 8: AMR-WB_Mode 12.65 kbit/s Bit 9: AMR-WB_Mode 8.85 kbit/s Bit 10: AMR-WB_Mode 6.60 kbit/s
Bit 11	"0"	normal IS-Message Sync Bit, constant
Bit 12...14	WB_MACS	The maximally supported number of Codec_Modes in this radio leg. Coding: "0.0.1" 1 Mode "0.1.0" 2 Modes "0.1.1" 3 Modes "1.0.0" 4 Modes "1.0.1" 5 Modes "1.1.0" 6 Modes "1.1.1" 7 Modes "0.0.0" 8 Modes
Bit 15	WB_Ver	Version Number of the AMR-WB TFO Scheme. Bit 15 is equivalent to the ATVN in Configuration Frames, see Annex C
Bit 16..18	CRC	3 CRC bits protecting Bits 2 to 10 and 12 to 15
Bit 19 20	EX	The normal 2 bits for IS_Message Extension: 00: No other extension block follows 11: An other extension block follows

7.11.3.1.3 AMR specific Codec_Attribute_Head Extension_Block

The AMR specific Codec_Attribute_Head Extension_Block (Table 7.11.3.1.3-1) shall precede the Codec Attribute Extension_Blocks of any AMR Codec_Type.

If PAR_Sel is set to "0" then one of 16 possible AMR Configurations is indicated in the PAR field and no additional Codec Attribute Extension_Blocks do follow.

Coding for PAR (bits 12.13.14.15):

0000: AMR_ACS with 10.2 / 6.70 / 5.90 / 4.75, NB_FS set to "0" and OM set to "0".

0001: AMR_ACS with 10.2 / 6.70 / 5.90 / 4.75, NB_FS set to "0" and OM set to "1".

0010: AMR_ACS with 7.4 / 6.70 / 5.90 / 4.75, NB_FS set to "0" and OM set to "0".
0011: AMR_ACS with 7.4 / 6.70 / 5.90 / 4.75, NB_FS set to "0" and OM set to "1".
other: reserved for future use.

If PAR_Sel is set to "1" then the AMR_ACS and potentially AMR_SCS is/are following.

Table 7.11.3.1.3-1: AMR specific Codec_Attribute_Head Extension_Block

Bit	Description	Comment
Bit 1	"0"	normal IS-Message Sync Bit, constant.
Bit 2	PAR_Sel	Differentiates this Extension_Block 0: Parameters included in PAR field: Simple Codec_List_Extension 1: Length Indicator (LI) included: Parameters follow in subsequent Extension_Blocks
Bit 3..10	CoID = HR_AMR FR_AMR UMTS_AMR UMTS_AMR2 OHR_AMR	This field identifies the AMR Codec_Type for which the subsequent attributes are valid. The same coding as in the Codec_x Extension_Block is used (long form)
Bit 11	"0"	normal IS-Message Sync Bit, constant
Bit 12.. 15:	LI / PAR	If Par_Sel==1: LI: Length Indicator: 0000: reserved; 0001: one other Extension_Block follows, etc. If Par_Sel==0: PAR: Codec specific definition of these four bits
Bit 16..18:	CRC	3 CRC bits protecting Bits 2 to 10 and 12 to 15
Bit 19..20:	EX	The normal 2 bits for IS_Message Extension: 00: No other extension block follows 11: An other extension block follows

7.11.3.1.4 AMR-WB SPECIFIC CODEC_ATTRIBUTE_HEAD EXTENSION_BLOCK

The AMR-WB specific Codec_Attribute_Head Extension_Block (Table 7.11.3.1.4-1) shall precede the Codec Attribute Extension_Blocks of any AMR-WB Codec_Type.

If PAR_Sel is set to "0" then one of 16 possible AMR-WB Configurations is indicated in the PAR field and no additional Codec Attribute Extension_Blocks do follow.Coding for PAR (bits 12.13.14.15):

0000: AMR-WB_ACS with 12.65 / 8.85 / 6.60.

0001: AMR-WB_ACS with 15.85 / 12.65 / 8.85 / 6.60.

0010: AMR-WB_ACS with 23.85 / 12.65 / 8.85 / 6.60.

other: reserved for future use.

If PAR_Sel is set to "1" then the AMR-WB_ACS and potentially AMR-WB_SCS is/are following.

Table 7.11.3.1.4-1: AMR-WB specific Codec_Attribute_Head Extension_Block

Bit	Description	Comment
Bit 1	"0"	normal IS-Message Sync Bit, constant.
Bit 2	PAR_Sel	Differentiates this Extension_Block 0: Parameters included in PAR field: Simple Codec_List_Extension 1: Length Indicator (LI) included: Parameters follow in subsequent Extension_Blocks
Bit 3..10	CoID = FR_AMR-WB UMTS_AMR-WB OHR_AMR-WB OFR_AMR-WB	This field identifies the AMR-WB Codec_Type for which the subsequent attributes are valid. The same coding as in the Codec_x Extension_Block is used (long form)
Bit 11	"0"	normal IS-Message Sync Bit, constant
Bit 12.. 15:	LI / PAR	If Par_Sel==1: LI: Length Indicator: 0000: reserved; 0001: one other Extension_Block follows, etc. If Par_Sel==0: PAR: Codec specific definition of these four bits
Bit 16..18:	CRC	3 CRC bits protecting Bits 2 to 10 and 12 to 15
Bit 19..20:	EX	The normal 2 bits for IS_Message Extension: 00: No other extension block follows 11: An other extension block follows

CLAIMS

1. Method for set up a tandem free operation in a communication network for speech communication between a first communication terminal (local MS) and a second communication terminal (distant MS), whereby at least one of the terminals uses at least one codec type to encode the speech signals into an encoded data representation, with a first transcoder (TRAU) and a second transcoder (TC) wherein messages are send from the first transcoder to the second transcoder and vice versa to determine if both communication terminals (local MS, distant MS) have at least one codec type in common and if this is the case to establish a data connection between the first communication terminal and the second communication terminal without having the need to insert transcoding functions into the signal path between the first and the second communication terminal comprising the steps of:
   - exchanging messages (TFO_REQ) between the transcoders that contain information (LUC;DUC) on the encoder type currently used by the communication terminals,
   - exchanging a second message (TFO_ACK) between the transcoders as a response to the first message (TFO_REQ) if both reported codec types match

characterized in that the first message contains further information (Ver; AMR_WB) on encoding capabilities of the respective communication terminal.

2. Method for set-up a transcoder free operation according to claim 1, wherein that further information relates to the version (Ver) of the transcoder free protocol version that is supported by the transcoder (TRAU; TC).

3. Method for set-up a transcoder free operation according to claim 1 or 2, wherein that further information comprises a list of additional codec types that are alternatively supported by the respective communication terminal.

4. Method for set-up a transcoder free operation according to claim 1, 2 or 3, wherein the further information comprises an indication that the message contains further information.

5. Method for set-up a transcoder free operation according to claim 2 wherein the version number is used in the receiving transcoder to look up a subset of an active codec set that is mandatory supported in each specific protocol version, wherby the transcoders compare that subset with the coder types supported by their mobile terminal and that the best codec type in common is chosen to enter into tandem free operation.

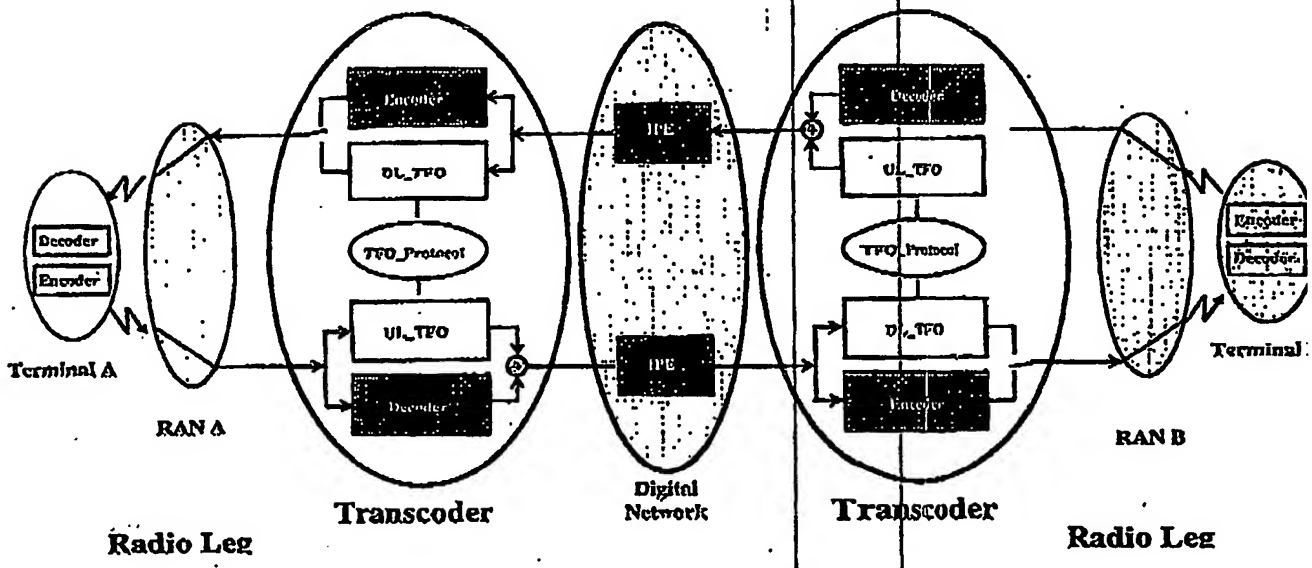
Decoder
Encoder
Terminal A
RAN A
Encoder
UL_TFO
TFO_Protocol
DL_TFO
Decoder
Transcoder
IPE
IPE
Digital Network
Decoder
UL_TFO
TFO_Protocol
DL_TFO
Encoder
Transcoder
RAN B
Encoder
Decoder
Terminal B
Radio Leg
Radio Leg

Fig. 1

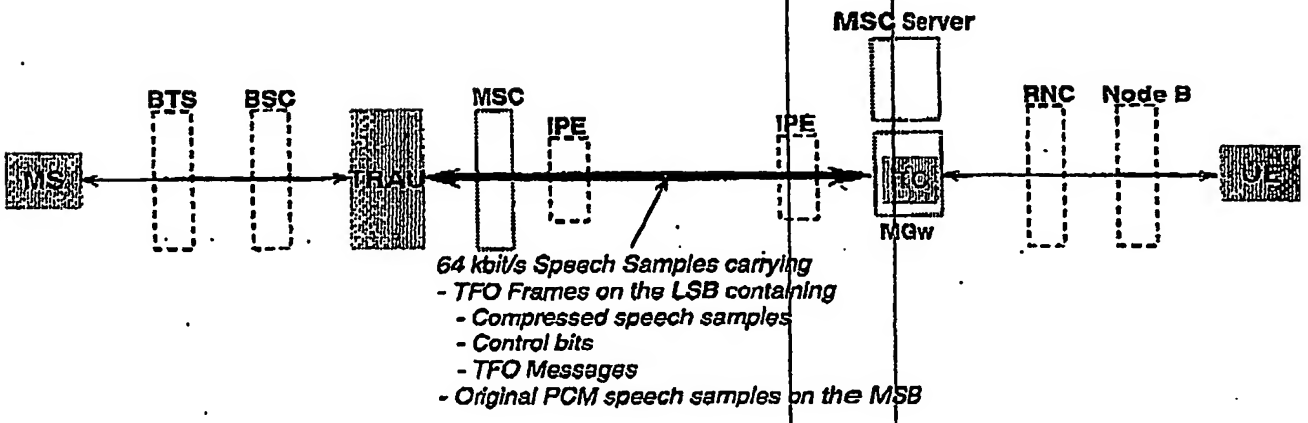
MSC Server
BTS
BSC
MSC
IPE
IPE
RNC
Node B
MS
TRAU
MGw
UE
64 kbit/s Speech Samples carrying
- TFO Frames on the LSB containing
- Compressed speech samples
- Control bits
- TFO Messages
- Original PCM speech samples on the MSB

Fig. 2

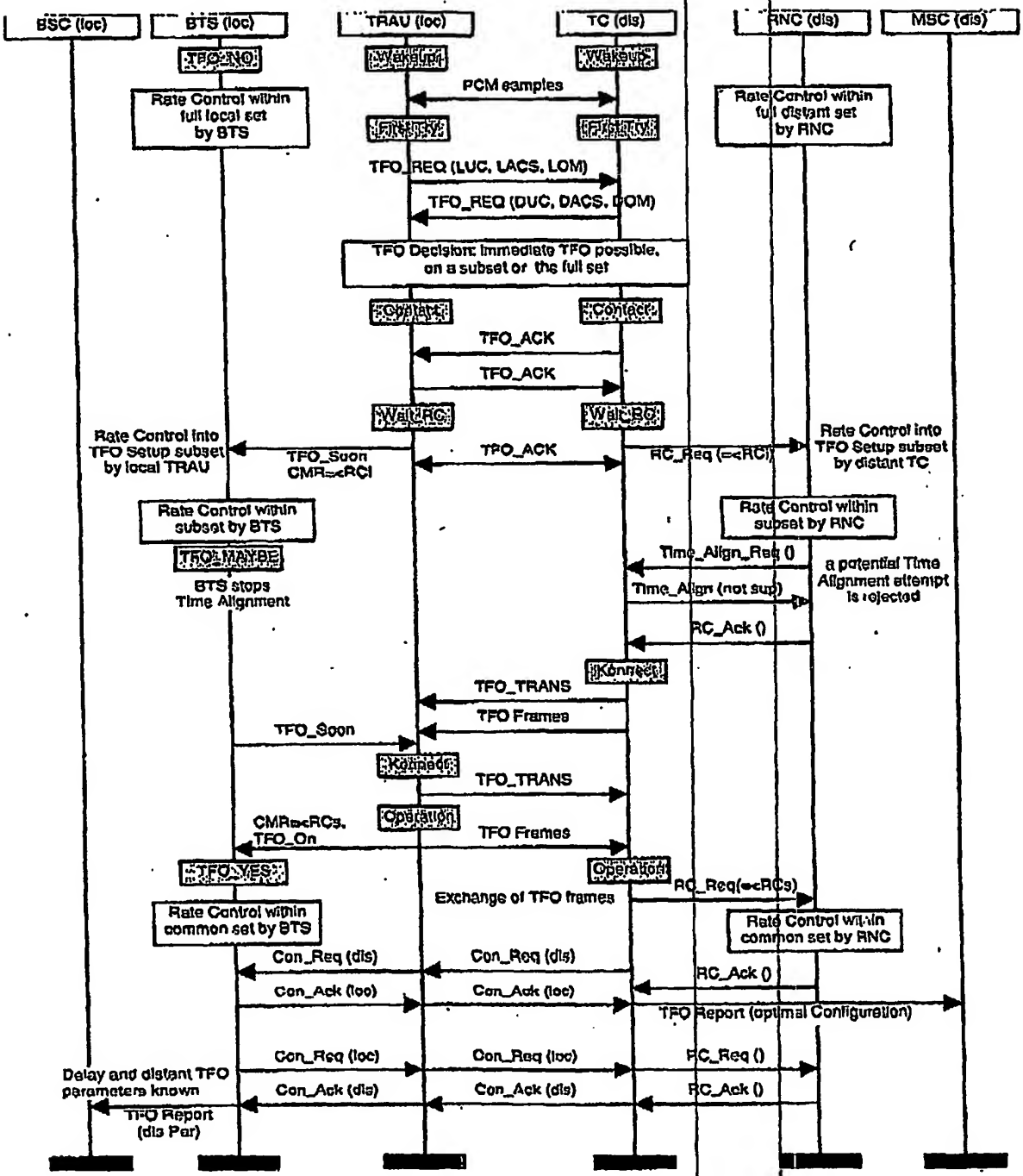
BSC (loc)
BTS (loc)
TRAU (loc)
TC (dis)
RNC (dis)
MSC (dis)
TFO_NO
Wakeup
Wakeup
PCM samples
Rate Control within full local set by BTS
Rate Control within full distant set by RNC
TFO_REQ (LUC, LACS, LOM)
TFO_REQ (DUC, DACS, DOM)
TFO Decision: Immediate TFO possible, on a subset or the full set
Connect
Connect
TFO_ACK
TFO_ACK
Wait RC
Wait RC
Rate Control into TFO Setup subset by local TRAU
TFO_Soon CMR=<RCI
TFO_ACK
RC_Req (=<RCI)
Rate Control into TFO Setup subset by distant TC
Rate Control within subset by BTS
Rate Control within subset by RNC
TFO_MAYBE
Time_Align_Req ()
a potential Time Alignment attempt is rejected
BTS stops Time Alignment
Time_Align (not sup)
RC_Ack ()
Konnect
TFO_TRANS
TFO Frames
TFO_Soon
Konnekt
TFO_TRANS
Operation
CMR=<RCs, TFO_On
TFO Frames
TFO_YES
Operation
RC_Req(=<RCs)
Exchange of TFO frames
Rate Control within common set by BTS
Rate Control within common set by RNC
Con_Req (dis)
Con_Req (dis)
RC_Ack ()
Con_Ack (loc)
Con_Ack (loc)
TFO Report (optimal Configuration)
Con_Req (loc)
Con_Req (loc)
RC_Req ()
Delay and distant TFO parameters known
Con_Ack (dis)
Con_Ack (dis)
RC_Ack ()
TFO Report (dis Par)

Fig. 3

3/3

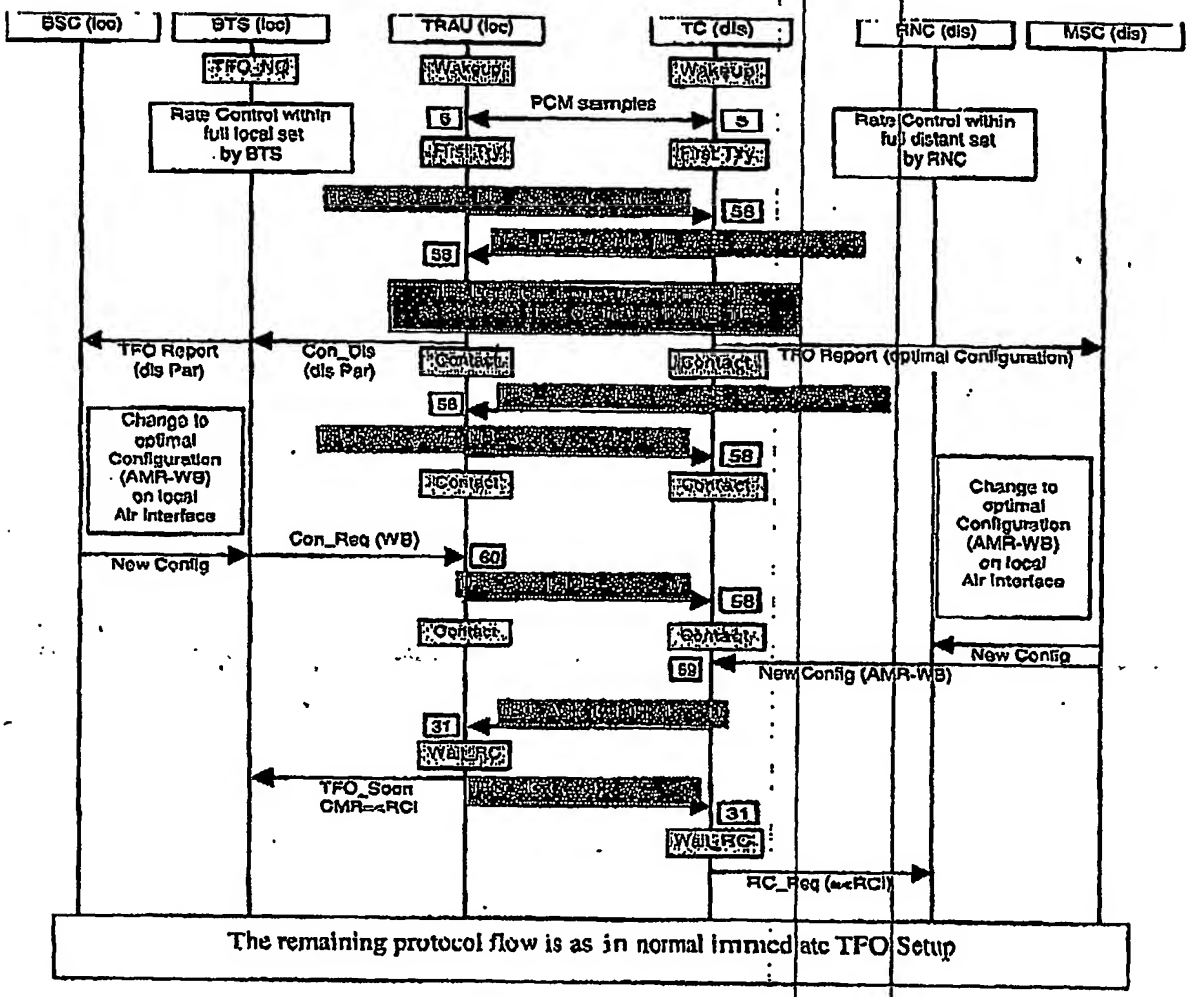
BSC (loc)
BTS (loc)
TRAU (loc)
TC (dis)
RNC (dis)
MSC (dis)
TFO_NG
WakeUp
WakeUp
PCM samples
5
5
Rate Control within full local set by BTS
Rate Control within full distant set by RNC
58
58
TFO Report (dis Par)
Con_Dis (dis Par)
Connect
Connect
TFO Report (optimal Configuration)
58
Change to optimal Configuration (AMR-WB) on local Air Interface
58
Connect
Connect
Change to optimal Configuration (AMR-WB) on local Air Interface
Con_Req (WB)
New Config
60
58
Connect
Connect
New Config
69
New Config (AMR-WB)
31
Wait RC
TFO_Scan CMR=<RCI
31
Wait RC
RC_Req (=<RCI)
The remaining protocol flow is as in normal immediate TFO Setup

Fig. 4

ABSTRACT

A Method for set up a tandem free operation in a communication network for speech communication between a first communication terminal (local MS) and a second communication terminal (distant MS) is discribed, whereby at least one of the terminals uses at least one codec type to encode the speech signals into an encoded data representation, with a first transcoder (TRAU) and a second transcoder (TC) wherein messages are send from the first transcoder to the second transcoder and vice versa to determine if both communication terminals (local MS, distant MS) have at least one codec type in common and if this is the case to establish a data connection between the first communication terminal and the second communication terminal without having the need to insert transcoding functions into the signal path between the first and the second communication terminal. In order to provide a very early possibility to establish a tandem free operation with the best codec type common to both mobile terminals it is proposed that the first message contains further information (Ver; AMR_WB) on encoding capabilities of the respective communication terminal.

Fig. 4